International Journal of Innovative Technology and Exploring Engineering (IJITEE)
ISSN: 2278-3075 (Online), Volume-1 Issue-5, October 2012

Two Dimensional Cellular Automata for Pseudo Random Number Generation

Vivekananda M, R.P.Das, K.V.Ramana Rao

Abstract*: *Cellular automata (CA) were initiated in the early 1950s to develop complex structures of capable self-reproduction and self-repair. Since then many researchers have taken attend in the study of CA. Initially, CA concept was first introduced by von Neumann von Neumann (1966) for the proposal of modeling biological self-reproduction. His primary interest was to derive a computationally universal cellular space with self-reproduction configurations. Afterward, a new phase of activities was started by Wolfram Wolfram (1983; 1984), who pioneered the investigation of CA as a mathematical model for self-organizing statistical systems. Wolfram was proved that the randomness of the patterns generated by maximum-length CA is significantly better than other widely used methods, such as linear feedback shift registers. The intensive interest in this field can be attributed to the phenomenal growth of the VLSI technology that permits cost-effective realization of the simple structure of local-neighborhood CA Wolfram (1986).

In this paper, an efficient PRNG based on hybrid between one-dimension (1-D) and two-dimension (2-D) CA is proposed. In the phase of the evolution of 2-D CA cells, the proposed CA PRNG is based on von Neumann neighborhood, in that this method refers to the five cells and new control values (rule decision input value & linear control input value) to decide a rule. In the phase of the evolution of 1-D CA cells, on the other hand,< 90, 150 > rule combination is used because this rule combination is better than the others Hortensius et al. (1989). In the meantime, the proposed CA PRNG is compared with previous works Guan et al. (2004); Tomassini et al. (2000) to check the quality of randomness. The proposed CA PRNG could generate a good quality of randomness because the proposed CA PRNG is better than the previous works and passed by the ENT Walker. In this project we have tried to verify the basic characterization of 2-D CA using a Xilinx Spartan 3E fpga.

Index Terms*: *About four key words or phrases in alphabetical order, separated by commas.

I. ONE-DIMENSION CA

A CA is a dynamical system in which space and time are discrete. A CA consists of an array of cells, each of which can be in one of a finite number of possible states, updated synchronously in discrete time steps, according to a local identical interaction, a rule. The next state of a cell is assumed to depend on itself and on its neighbors. The cells evolve in discrete time steps according to some deterministic rule that depends only on local neighbors. The next-state function for a three-neighborhood CA cell in one-dimension can be expressed as following equation (1)

$$s_i(t+1) = f(s_{i-1}(t),\ s_i(t),\ s_{i+1}(t)) \qquad (1)$$

where f denotes the local function realized with a combinational logic, such as AND, OR, XOR, and NOT,i is the position of an individual in the one-dimensional array of the cell, t is the time step, s I (t) is the output state of the i th cell at the t th time step, and s I (t + 1) is the output state of thei th cell at the (t+1) th time step. In a two-state, three-neighborhood CA there can be a total of 2^3 (from 000 to 111) distinct neighborhood configurations. For such a CA there can be a total of(2 $^{2) \, 3}$ (= 256) distinct mappings from all these neighborhood configurations to the next state. Each mapping is called a rule of CA. If the next-state function of a cell is expressed, then the decimal equivalent of the output is conventionally called the rule number for the cell Wolfram (1986).

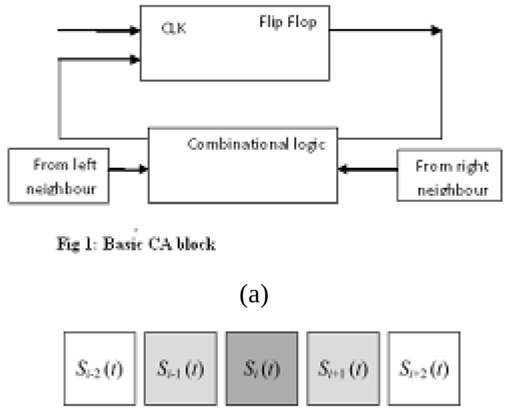


Fig 1: Basic CA block

(a)

(b)

Fig. 1. A CA cell structures: (a) a 2-state, 3-neighborhood CA cell, (b) an example of CA cells with 3-neighborhood in 1-D

1D Cellular Automata the CA structure investigated by Wolfram can be viewed as discrete lattice of sites (cells) where each cell can assume either the value 0 or 1. The next state of a cell is assumed to depend on itself and on its two neighboring cells for a 3 neighborhood dependency. The cells evolve in discrete time steps according to some deterministic rule that depends only on local neighborhood. In effect, each cell as shown in Fig 1, consists of a storage element (D- Flip Flop) and a combinational logic implementing the next state.

If the next state f unction of a cell is expressed in the form of a truth table, then the decimal equivalent of the output is conventionally called the rule number for the cell. Thus for a 3 neighborhood CA, the next state function for cell i is represented as follows for each of 256 rules. The top rows gives all the possible states of the neighborhood cells at the time instant (t), while the 2 nd and 3 rd rows give the corresponding states of the ஂ க□ cell at the time instant (t+1) for two illustrative CA rules.

Revised Manuscript Received on 30 October 2012
***Correspondence Author(s)

M.Vivekanada, ECE Department,JNTUK University/Pydah College of Engineering and Technology/ Visakhapatnam, India.

R.P.Das, ECE Department, JNTUK University/ Pydah College of Engineering and Technology / Visakhapatnam, India.

The Second Row taken as binary number and converted into decimal representation is the rule no. 90. Similarly, the third row corresponds to rule no. 150. The expression for a rule can be obtained from its truth table. The minimized expression for rule 90 & rule 150 is

$$q_i(t+1) = q_{i-1}(t) \oplus q_{t+1}(t)$$
$$q_i(t+1) = q_{i-1}(t) \oplus q_i(t) \oplus q_{t+1}(t)$$

Definitions:

1. If same rule is applied to all the cells in a CA, then the CA is said to be Uniform or Regular CA.
2. If different rules are applied to different cells in a CA, then the CA is said to be Hybrid CA.
3. The CA is said to be a Periodic CA if the extreme cells are adjacent to each other.
4. The CA is said to be a Null boundary CA if the extreme cells are connected to logic 0-state.
5. If in CA the neighbourhood dependence is on XOR or XNOR only, then the CA is called additive CA, specifically, a linear CA employs XOR rules only.
6. A CA whose transformation is invertible (i.e. all the states in the state transition diagram lie in some cycle) is called a Group CA, otherwise it is called a Non Group CA.

2D Cellular automata.

A two-dimension (2-D) CA is a generalization of a 1-D CA, where the cells are arranged in a two-dimensional grid with connections among the neighboring cells. For a 2-D CA, types of cellular neighborhoods are usually considered: five cells, consisting of one cell along with its four immediate non-diagonal neighbors (also known as the von Neumann neighborhood); and nine cells, consisting of one cell along with its eight surrounding neighbors (also known as the Moore neighborhood). In this paper, a type of von Neumann neighborhood is used which considers five-neighborhoods that these consist ofself, top, bottom , left and right.

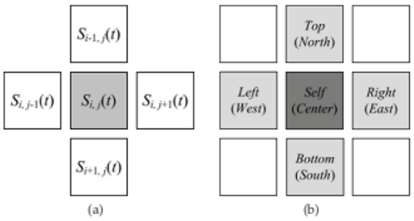


Fig. 3. The geometric representation of the von Neumann neighborhood.

Figure 3 shows the type of von Neumann neighborhood with a five-neighborhood dependency.

The next states i , j (t + 1) of a 2-D CA is given by

$$s_{i,j}(t+1) = f(s_{i-1,j}(t), s_{i,j-1}(t), s_{i,j}(t), s_{i,j+1}(t), s_{i+1,j}(t)).$$

Since f is a Boolean function of five variables, there are 2 5 = 32 distinct neighborhood configurations. Hence to express a transition rule of a 2-D CA in a manner similar to a 1-D CA, 32-bits are considered which is almost impossible to manage practically. In 2D cellular automata the cells are arranged in two dimensional grid with connections among the neighborhood cells. The state of CA at any time instan can be represented by an ଁ ୧୧ଃ ଁ binary matrix. The neighbourhood function specifying the next state of a particular cellof the CA is affected by the current state of itself and eight cells in its nearest neighborhood. Mathematically, the next state q of the cell of a 2D CA given by

$$q_{ij}(t+1) = f[q_{i-1,j-1}(t), q_{i-1,j}(t), q_{i-1,j+1}(t), q_{i,j-1}(t), q_{i,j+1}(t), q_{i+1,j-1}(t), q_{i+1,j}(t), q_{i+1,j+1}(t)\ ]$$

Where is the Boolean function of 9 variables? To express a transition rule of 2D CA, a specific rule convention proposed in [17] is noted below.

64	128	256
32	1	2
16	8	4

The Central box represents the current cell (that is the cell being considered) and all other boxes represent the eight nearest neighbors of that cell. The number within each box represents the rule number associated with that particular neighbor of the current cell – that is, if the next state of a cell is dependent only on its present state, it is referred to as rule 1. If the next state depends on the present state of itself and its right neighbor, it is referred to as rule 3(=1+2). If the next state depends on the present state of itself and its right, bottom, left, and top neighbors, it is referred to as rule 171 (=1+2+8+32+128) and so on.

II. MATHEMATICAL MODEL

This 2D CA behavior can be analyzed with the help of an elegant mathematical model where we use two fundamental matrices to obtain row and column dependencies of the cells. Let the two dimensional binary information matrix be denoted as Xt that represents the current state of a 2D CA configured with a specific rule. The next state of any cell will be obtained by XOR operation of the states of its relevant neighbors associated with the rule. The global transformation associated with different rules can be made effective with following fundamental matrices referred to as T1 and T2 in the rest of the paper.

$$T_{1=}\begin{pmatrix}0 & 1 & 0\\ 0 & 0 & 1\\ 0 & 0 & 0\end{pmatrix} \text{ and } T_2 = \begin{pmatrix}0 & 0 & 0\\ 1 & 0 & 0\\ 0 & 1 & 0\end{pmatrix}$$

The following lemma specifies the value of the next state of a 2D CA referred to as given that its current state is. The CA is assumed to be configured with primary rule only—that is it depends on only one of its nine neighbors. Lemma: 1. The next state transition of all the primary rules (1, 2, 4, 8, 16, 32, 64, 128, 256) with null boundary condition, can be represented as

$$\begin{array}{lll}
Rule\ 1 & \rightarrow [X_{t+1}] & = [X_t] \\
Rule\ 2 & \rightarrow [X_{t+1}] & = [X_t][T_2] \\
Rule\ 4 & \rightarrow [X_{t+1}] & = [T_1][X_t][T_2] \\
Rule\ 8 & \rightarrow [X_{t+1}] & = [T_1][X_t] \\
Rule\ 16 & \rightarrow [X_{t+1}] & = [T_1][X_t][T_1] \\
Rule\ 32 & \rightarrow [X_{t+1}] & = [X_t]\ [T_1] \\
Rule\ 64 & \rightarrow [X_{t+1}] & = [T_2][X_t][T_1] \\
Rule\ 128 & \rightarrow [X_{t+1}] & = [T_2][X_t] \\
Rule\ 256 & \rightarrow [X_{t+1}] & = [T_2][X_t][T_2]
\end{array}$$

Lemma: 2. The next state transition of all configured with a secondary rule can be represented as modulo 2 sum of matrices of the concerned primary rules

For Example:

1) Rule 3 = Rule 1 + Rule 2, the next state transition can be repr esented as

Published By:
Blue Eyes Intelligence Engineering
and Sciences Publication (BEIESP)
© Copyright: All rights reserved.

$$[X_{t+1}] \quad = [X_t] + [X_t][T_2]$$

2) Rule 170 = Rule 2 + Rule 8 + Rule 32 + Rule 128, the next state transition for rule 170 can be represented as

$$\begin{aligned}&[X_{t+1}]\\&= [X_t][T_2] + [T_1][X_t][X_t]\,[T_1] + [T_2][X_t]\\&= [X_t][T_1 + T_2] + [T_1 + T_2][X_t]\\&= [X_t][s] + [s][X_t]\end{aligned}$$

Where $[s] = [T_1 + T_2]$

Lemma: 3. The equivalent one dimensional map matrix for any rule R can be represented as

$$[T_R]_{mn\,x\,mn} = \begin{bmatrix} D & U & 0 & & 0 & 0 & 0\\ L & D & U & \cdots & 0 & 0 & 0\\ 0 & L & D & & 0 & 0 & 0\\ & \vdots & & \ddots & & \vdots & \\ 0 & 0 & 0 & \cdots & L & D & U\\ 0 & 0 & 0 & & 0 & L & D\end{bmatrix}$$

Where D, L and U are one of the following matrices of the order of n x n:

$$[0],\ [I],\ [T_1],\ [T_2], [I \quad + T_1], [I + T_2], [S] and\ [I + S]$$

Lemma: 4. Rank of fundamental matrices

$$(T_1)_{n\ x\ n}\ and\ (T_2)_{n\ xn}\ \text{is}\ n - 1.$$

Theorem: 1. All primary rules other than rule 1, with null boundary condition, are non group CA Lemma: 5. The next state transition of all the primary rules with periodic boundary condition can be represented by

$$\begin{aligned}&Rule\ 1 && \rightarrow [X_{t+1}] && = [X_t]\\&Rule\ 2 && \rightarrow [X_{t+1}] && = [X_t][T_{2c}]\\&Rule\ 4 && \rightarrow [X_{t+1}] && = [T_{1c}][X_t][T_{2c}]\\&Rule\ 8 && \rightarrow [X_{t+1}] && = [T_{1c}][X_t]\\&Rule\ 16 && \rightarrow [X_{t+1}] && = [T_{1c}][X_t][T_{1c}]\\&Rule\ 32 && \rightarrow [X_{t+1}] && = [X_t]\,[T_{1c}]\\&Rule\ 64 && \rightarrow [X_{t+1}] && = [T_{2c}][X_t][T_{1c}]\\&Rule\ 128 && \rightarrow [X_{t+1}] && = [T_{2c}][X_t]\\&Rule\ 256 && \rightarrow [X_{t+1}] && = [T_{2c}][X_t][T_{2c}]\end{aligned}$$

Where

$$T_{1c=} \begin{pmatrix} 0 & 1 & 0\\ 0 & 0 & 1\\ 1 & 0 & 0\end{pmatrix} \quad \text{and}$$

$$T_{2c} = \begin{pmatrix} 0 & 0 & 1\\ 1 & 0 & 0\\ 0 & 1 & 0\end{pmatrix}$$

Lemma: 6. The Matrix for any periodic boundary CA rule (PR) can be represented as

$$[T_{PR}]_{mn\,x\,mn} = \begin{bmatrix} D & U & 0 & & 0 & 0 & U_c\\ L & D & U & \cdots & 0 & 0 & 0\\ 0 & L & D & & 0 & 0 & 0\\ & \vdots & & \ddots & & \vdots & \\ 0 & 0 & 0 & \cdots & L & D & U\\ L_C & 0 & 0 & & 0 & L & D\end{bmatrix}$$

Where D, L, U, Lc and Uc are one of the following matrices of the order of nxn:

$$[0],\ [I],\ [T_{1c}],\ [T_{2c}], [I \quad + T_{1c}], [I + T_{2c}], [S_C] and\ [I + S_C]$$

Theorem 2: All primary rules with periodic boundary condition, are group rules.

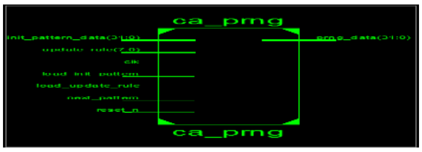
III. IMPLEMENTATION AND RESULTS

We have implemented the module by adopting the properties of Xilinx xc3s200 fpga and the design summary was given below.

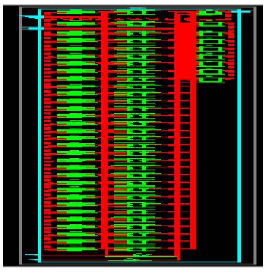
IV. DESIGN SUMMARY

Device Utilization Summary (estimated values)			[-]
Logic Utilization	**Used**	**Available**	**Utilization**
Number of Slices	92	1920	4%
Number of Slice Flip Flops	48	3840	1%
Number of 4 input LUTs	162	3840	4%
Number of bonded IOBs	77	173	44%
Number of GCLKs	1	8	12%

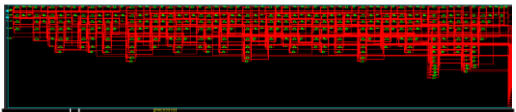
RTL MODULE

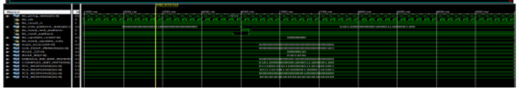


RTL SCHEMATIC



TECHNOLOGY SCHEMATIC





OUTPUT WAVE FORM

REFERENCES

1. G. O. Young, "Synthetic structure of industrial plastics (Book style with paper title and editor)," in *Plastics*, 2nd ed. vol. 3, J. Peters, Ed. New York: McGraw-Hill, 1964, pp. 15–64.
2. W.-K. Chen, *Linear Networks and Systems* (Book style). Belmont, CA: Wadsworth, 1993, pp. 123–135.
3. H. Poor, *An Introduction to Signal Detection and Estimation*. New York: Springer-Verlag, 1985, ch. 4.
4. B. Smith, "An approach to graphs of linear forms (Unpublished work style)," unpublished.
5. E. H. Miller, "A note on reflector arrays (Periodical style—Accepted for publication)," *IEEE Trans. Antennas Propagat.*, to be published.
6. J. Wang, "Fundamentals of erbium-doped fiber amplifiers arrays (Periodical style—Submitted for publication)," *IEEE J. Quantum Electron.*, submitted for publication.
7. C. J. Kaufman, Rocky Mountain Research Lab., Boulder, CO, private communication, May 1995.
8. Y. Yorozu, M. Hirano, K. Oka, and Y. Tagawa, "Electron spectroscopy studies on magneto-optical media and plastic substrate interfaces(Translation Journals style)," *IEEE Transl. J. Magn.Jpn.*, vol. 2, Aug. 1987, pp. 740–741 [*Dig. 9th Annu. Conf. Magnetics* Japan, 1982, p. 301].
9. M. Young, *The Techincal Writers Handbook.* Mill Valley, CA: University Science, 1989.
10. (Basic Book/Monograph Online Sources) J. K. Author. (year, month, day). *Title* (edition) [Type of medium]. Volume(issue). Available: http://www.(URL)
11. J. Jones. (1991, May 10). Networks (2nd ed.) [Online]. Available: http://www.atm.com
12. (Journal Online Sources style) K. Author. (year, month). Title. *Journal* [Type of medium]. Volume(issue), paging if given. Available: http://www.(URL)
13. R. J. Vidmar. (1992, August). On the use of atmospheric plasmas as electromagnetic reflectors. *IEEE Trans. Plasma Sci.* [Online]. *21(3).* pp. 876—880. vailable:http://www.halcyon.com/pub/journals/21ps03-vidmar